A Computer Vision Integration Model for a Multi-modal Cognitive System

Vrecko, A; Skocaj, D; Hawes, Nicholas; Leonardis, A

DOI:
10.1109/IROS.2009.5354358

A Computer Vision Integration Model for a Multi-modal Cognitive System

Alen Vrečko, Danijel Skočaj, Nick Hawes and Aleš Leonardis

***Abstract*— We present a general method for integrating visual components into a multi-modal cognitive system. The integration is very generic and can work with an arbitrary set of modalities. We illustrate our integration approach with a specific instantiation of the architecture schema that focuses on integration of vision and language: a cognitive system able to collaborate with a human, learn and display some understanding of its surroundings. As examples of cross-modal interaction we describe mechanisms for clarification and visual learning.**

I. Introduction

Computer vision methods are often researched and developed in isolation, and evaluated according to human visual criteria — they are expected to emulate our perception of the world. Many methods achieve excellent results on standard benchmark image databases [1], however they often fail to achieve a similar performance within real systems that are supposed to operate in real world and in real time. Not so rarely our expectations toward isolated visual systems reflect higher levels of our visual cognition, which often exceeds their actual scope. In some cases the excessive expectations can hinder the development of otherwise promising methods.

Multi-modal cognitive systems represent a different challenge for computer vision methods. In this case the main and most direct benchmark for a visual component's performance is the benefit for other system components, which is eventually reflected in the performance of the cognitive system as a whole. The multi-modality of the integrated systems ensures that all its components are exposed (through cross-modal communication) to a much wider spectrum of environmental input than if they were working in isolation. This usually increases their complexity and development effort on the one hand, but improves their reliability on the other, since they can benefit from other components' output. Our expectations are in this case focused to the system level, which makes it easier for the vision component development to concentrate on intra-modal and cross-modal communication.

The availability of cross-modal information can be a crucial advantage for any visual component. It can be used for verifying component's decisions or even as an important cue when facing more than one hypothesis of similar probabilities. If the system is able to learn (we firmly believe that learning ability is one of the most important requirements for a cognitive system), the cross-modal information can be used for labeling visual information or for detecting gaps in knowledge. Of course, to be able to exploit these benefits, the visual subsystem has to be capable to discern situations when no external information is needed from those that require cross-modal verification.

In this paper we introduce a novel approach to integration of visual components into a multi-modal cognitive system. The main advantage of our method is that it is very generic in the sense that it applies to a multitude of possible *intra-modal* (components that make up the visual subsystem), as well as *multi-modal* (subsystems composing cognitive system) instantiations.

The problem of the cross-modal vision integration is in its core a *symbol grounding problem*, introduced by Harnad in [8]. Similar problems have been very often addressed in the literature, e.g. by Chella *et al* [2], [5] and Roy [16], [15]. Our work differs from the work of the authors mentioned above in that we seek solutions within a much wider and general cognitive framework, which assumes also continuous and parallel execution. The integration of visual subsystem into the framework is very generic an can work with minimal modifications with an arbitrary set of other modalities, using high-level, a-modal entity representations. The visual-linguistic instantiation we show in this work is just one example of possible cross-modal combinations. In this sense a similar approach is followed in [3]. Focusing to visual-linguistic integration, we see the main advantage of our work in more diversified, autonomous and responsive cross-modal learning mechanisms (implicit and explicit learning, clarification).

In Section II we briefly introduce the architecture schema our system is built atop and give a general overview of the system. Section III describes the visual part of the system and its interaction with other modalities, while in Section IV we exemplify the cross-modal mechanisms shown in Section III. Finally, Section V summarizes and concludes the discussion.

II. System Overview

A. CoSy Architecture Schema

The integrated computer vision system we present in this work is built atop the *CoSy Architecture Schema (CAS)* [10], [9] implemented by *CAS Toolkit (CAST)* [11], [9], [6]. CAS divides the cognitive system into loosely coupled subarchitectures (SAs), where each of them roughly encompasses one *modality* (e.g. Vision SA, Communication SA, Manipulation SA, etc.) or wraps up some *a-modal* or

A. Vrečko, D. Skočaj and A. Leonardis are with the Faculty of Computer and Information Science, University of Ljubljana, Slovenia, {alen.vrecko, danijel.skocaj, ales.leonardis}@fri.uni-lj.si

N. Hawes is with the School of Computer Science, University of Birmingham, UK, n.a.hawes@cs.bham.ac.uk

mediative functionality (e.g. Binding SA, Motivation SA). Figure 1 shows the general SA layout.

The SAs consist of a set *of processing components* that are able to share information with each other. The communication is mostly indirect, following the well known 'blackboard' multi-agent communication approach [7]. The components can push their shareable data to a special component called *'Working Memory'* (WM), where it can be accessed by other components. Each SA has its own WM component, which is used by default by SA member components. Components can also access other SA's WMs, but this kind of communication (with the exception of mediative SA's WMs) should be avoided or held to its absolute necessary minimum. An alternate communication option is a direct link between components. Communication via working memory offers a high degree of flexibility and scalability to the system. The components can access the data in working memory without knowing its source. A component can thus just by monitoring the state of a single component access the information from multiple independent components. The direct communication approach is usually necessary for more efficient access to larger data structures or data streams (e.g. video streams). In this case the component has to know its data supplier. The direct communication is usually more likely the closer the component is to the sensorial data, while the blackboard data sharing is almost exclusive among the higher level components.

Another special SA component is the *Task Manager* that is used to coordinate actions between processing components. In this sense we divide the processing components in two types:

- The **Managed processing components** require the Task Manager's permission to execute their information processing tasks (e.g. to add, delete or change something in WM). Their actions are usually triggered by certain events in WM, therefore they are also called *event-driven* components.
- The **Unmanaged processing components** do not interact with the Task Manager, at all, while the interaction with the WM is limited: they can add new entries to WM, but they can not read anything from it, nor they are sensitive to WM events. Usually they are connected via a direct link to an external data stream (e.g. video stream), and writting the processing results to WM. Hence they are also called *data-driven* components.

B. CAS Instantiations

The flexibility and scalability of the architecture allows easy addition of new components to the system, enabling phased approach to the system development. The system described in this work is one of the possible instantiations of the architecture schema. Figure 2 gives an overview of a typical CAS instantiation. The system is composed of several SAs which operate in a distributed, asynchronous and collaborative way. In this paper we focus on the Vision SA, which is used to illustrate our general approach to the integration of visual components.

III. Visual Subarchitecture

The goal of the visual subsystem is to provide reliable visual information to other modalities that are part of the cognitive system. It consists of a set of relatively simple, but specialized components sharing information with each other. The Visual Subarchitecture can be divided into three layers (Figure 3):

- the lower, *quantitative layer* deals directly with the sensorial input and provides quantitative analysis of the scene as a whole,
- the middle, *qualitative layer* performs qualitative analysis of selected regions of the scene,
- the upper, *interface layer* exchanges information with other modalities.

Another goal we are trying to achieve with the distributed approach is improving the robustness of the system. Since we are aware that only a limited degree of robustness can be achieved on the component level, we try to compensate this on the integration level. An important quality that is therefore required from the components is the ability of self-evaluation. Since their output information is rarely the final system output, but is usually reused by other components (possibly from different modalities), they have to be able to determine the reliability of their processing results. Only reliable information should be available to other components. If this is not possible, the components should share partial processing results instead, or try to seek the missing information elsewhere. In this sense redundancy in information processing is not only desired, but often also required. Output information can also be expressed as probability distribution over several alternatives, postponing the selection to higher level processing, where more cross-modal information is available.

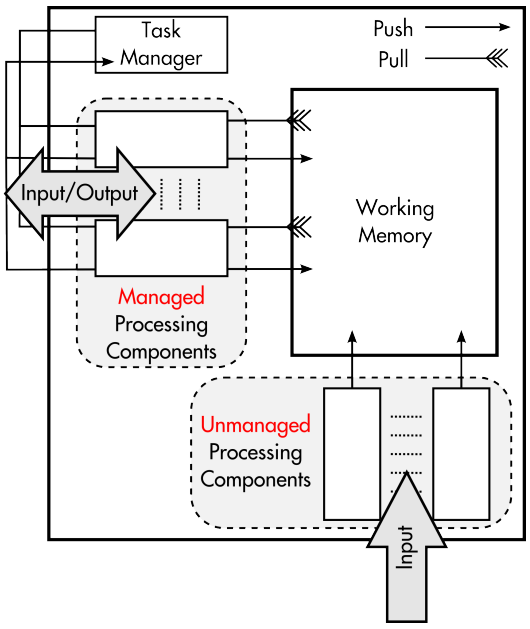

Fig. 1. The CAS Subarchitecture Design Schema. For more details consult [11], [9].

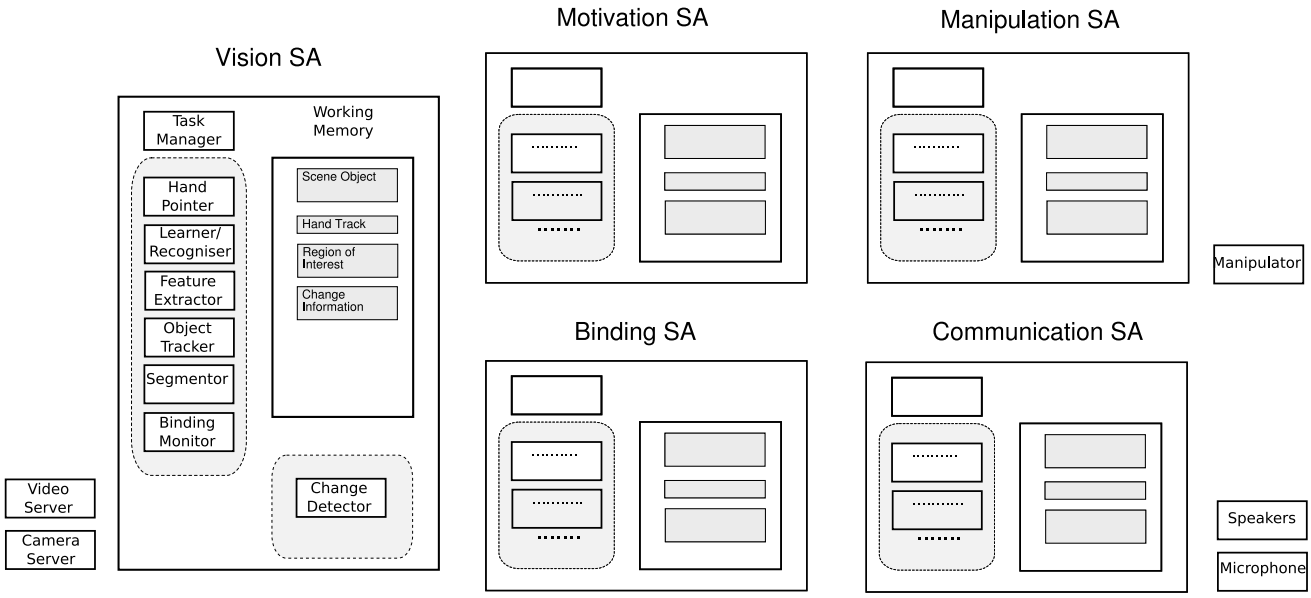

Fig. 2. A typical CAS instantiation consisting of Vision SA, Motivation SA, Binding SA, Manipulation SA (robot arm) and Communication SA (language).

A. Quantitative Layer

In our current instantiation the Vision SA uses a single sensory input from a static camera above the desktop surface. The video input is managed by the **Video Server** component, which provides access to videostream data through direct connection. Video frames are retrieved by four components:

- The **Change Detector** is a data-driven component that is sensitive to the changes in the scene. Whenever it detects a certain degree of activity within the scene, it notifies the interested components by updating a special data structure in the WM. This notification can be used by other components as a trigger for their processing. It sends a similar signal to the WM when the scene activity ceases.
- The **Segmentor** is a managed component that segments the video frames trying to determine *Regions of Interest (ROI)* in the scene. The segmentation takes place every time when the scene becomes static and is currently based on background subtraction. The component relies on information provided by Change Detector and Handpointing Detector about the activity in the scene. After the segmentation is done, the component tries to distinguish between newly segmented ROIs and those from the previous segmentation by simply matching their locations and areas. Based on ROI matching the component adds or deletes ROI representations in WM.
- The **Object Tracker** component is designed to follow moving objects in the scene. Tracking is based on the objects' color histograms that are retrieved from ROI data structures in WM. The tracking of an individual object is therefore triggered by the addition of its ROI to WM. The component is constantly updating the tracked objects' positions in WM ROI structures. This mechanism allows Segmentor to reidentify the moved objects' ROIs within the scene, rather than assert new ROIs for them.
- The **Handpointing Detector** is a means of visual communication with the human tutor. Based on the skin color segmentation and fast template matching the component is sensitive to the presence of human hand in the scene. In the case of a hand presence it tries to detect its index finger and determine the nearest object (ROI) in the pointing direction. The pointed object is deemed salient and is kept along with the hand status information in a WM data structure. While the hand is in scene, the component is overriding the static scene signal, so that the segmentation can not be performed,

The above five components form the lower, *quantitative layer* of the visual system, dealing with the quantitative scene information. In general, this level detects and monitors scene parts that might be interesting to higher level processing (the bottom-up attention mechanism). Similarly, the higher levels can provide a feedback about the types of information they are currently interested in (the top-down attention mechanism).

B. Qualitative Layer

The components of the quantitative layer are usually directly processing the sensorial input. Qualitative information about the individual scene objects is maintained by the middle, *qualitative layer*. In our current instantiation it is formed by two components:

- The **Feature Extractor's** task is to extract visual features from ROIs and update the WM ROI structures accordingly. The component could in principle handle any type of visual features. Currently the features include median HSV color components and several other shape features.

- The **Learner-recognizer** component maintains internal visual knowledge in the form of associations between low-level visual features and high-level cross-modal concepts (e.g. visual properties). The representation of each visual concept is based on *Kernel Density Estimator (KDE)* [14], [17] and can be *incrementally updated*. Visual properties represent basic descriptive object properties, like various colors and basic shapes (e.g. red, blue, green; circular, rectangular, triangular, etc.). The component uses the underlying KDE representations to determine object's properties based on extracted visual features. Object's properties are stored along with the other higher level object information in a separate WM structure (*'SceneObject'*).
The KDE based *recognition* is able to evaluate the reliability of its results. Only the information that exceeds reliability threshold is published in 'SceneObject' WM structure. In case of non-reliable recognition results, the SA has an option to request confirmation or clarification from other SAs.
An important part of component's activity is *visual learning*, which is achieved by updating the KDE representations. In order to perform an update of the current representations the components need extracted features and information about the object's properties. In general the information that can be translated to learnable visual properties can be provided by other components, usually from other modalities. In our case this information is supplied by the communication sub-system, which analyzes tutor's utterances. A more detailed description of cross-modal visual learning follows in Section III-C.2.
Our system has been designed to operate in a continuous way, therefore to keep continuously updating the knowledge (the current representations), possibly also in an unassisted way. In case of erroneous updates, this may lead to the propagation of errors and degrading the models. Therefore, the learning algorithms must have also the possibility to correct the current representations, thus to remove the erroneously incorporated information from the model. The KDE based learning methodology that we have developed supports such kind of *unlearning*[14], [17]. With unlearning the system can react to explicit cross-modal negative feedback information and can unlearn the representations of the corresponding concepts accordingly.

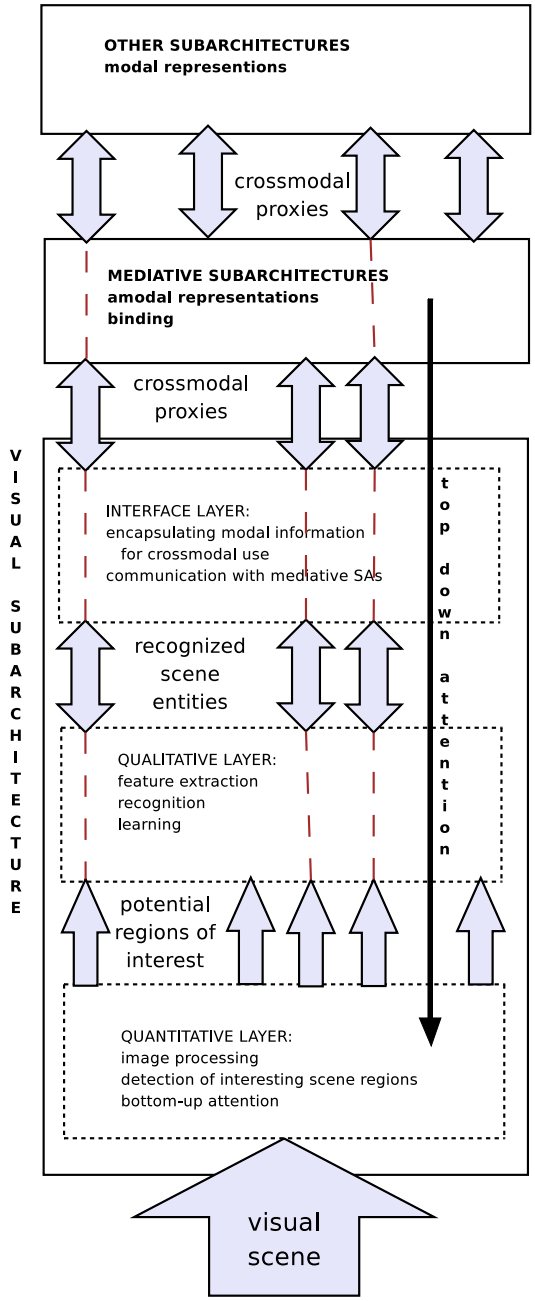

Fig. 3. The general layout of the Visual Subarchitecture and its cross-modal interaction. The red dashed lines span the representations of the same entities across different vision layers and modalities.

In general the qualitative layer processes the scene regions deemed interesting by the quantitative layer, looking for cues to decide if and in what way they map to some modal entity templates (e.g. scene objects). Once the entities are established, they are processed individually. Usually this involves the recognition of their properties. A desirable property of the components on this layer is the ability to evaluate the recognition confidence.

The upper tier of the system — the *interface layer* is composed of monitoring components. Their purpose is to exchange information with components from other modalities. The exchange of information is usually achieved via dedicated mediative subsystems. In contrast to modal SAs, the a-modal mediative SAs are known to all other SAs in the system. The monitors forward the selected data from local WM to those subsystems and make other cross-modal data available to local components. Our system currently has two such components

- The **Visual Binding Monitor**'s task is to exchange information with the *Binding Subarchitecture*. The basic principles of cross-modal information exchange via

Binding SA are explained in Section III-C.1. Typically the binding monitor maintains the cross-modal representations of recognized properties of currently perceived scene objects.

- The **Motivation Monitor** is in general used for forwarding component's requests to another mediative subsystem — the Motivation Subarchitecture. Requests in Motivation SA usually result in some additional processing in one or more SAs[1]. An example of such a request is the clarification request, which is the means for obtaining additional information about a scene object from other modalities. The retrieval of such information is not part of the routine SA's processing, therefore is not available in cross-modal representation of the scene objects by default.

C. Cross-modal Integration

1) Binding Subarchitecture: Individual modalities within the cognitive system form their own internal representations of the environment. An object on the table can appear as a segmented color patch to the visual system, a set of graspable areas to the robot arm touch sensors or a reference in a tutor's utterance to the communication subsystem. Each modality tries, based on its own innate and learnt concepts, to group the different sensorial cues into modal representations of distinct entities. The SA's binding monitor converts those representations to a set of *binding features* — a special representation form for cross-modal communication. Through the binding monitor each modal representation of each perceived entity delegates to the Binding SA its own representative — the *binding proxy*, containing its binding features. The role of the Binding SA is to group the corresponding proxies to *binding unions* by comparing their sets of binding features. Binding unions can be regarded as a-modal representations of perceived entities. Each entity can thus have several modal representations, but one a-modal representation, only. Binding unions are used as a sort of communication channels between different modalities, ensuring that the exchanged multi-modal information pertains to the same entity. The a-modal symbols are thus grounded in several modal representations and through them in sensorial information (see the dashed red lines in Figure 3).

A more detailed description of the binding process is available in [12], [13].

2) Cross-modal Visual Learning: As a part of a complex multi-modal cognitive architecture the visual subsystem is expected to process sensorial input in real time providing reliable information about the visual environment to other modalities. At the same time, the subsystem should be able to access and use to its advantage the information from other modalities. This information often includes the feedback to the system's past behavior (e.g. response to its past actions or previously forwarded information). In a continuous effort

[1]Motivation triggered processing involves also planning which is performed by the Planning SA, which is, however, beyond the scope of this paper. Consult [4] for a detailed description of this part of our cognitive system.

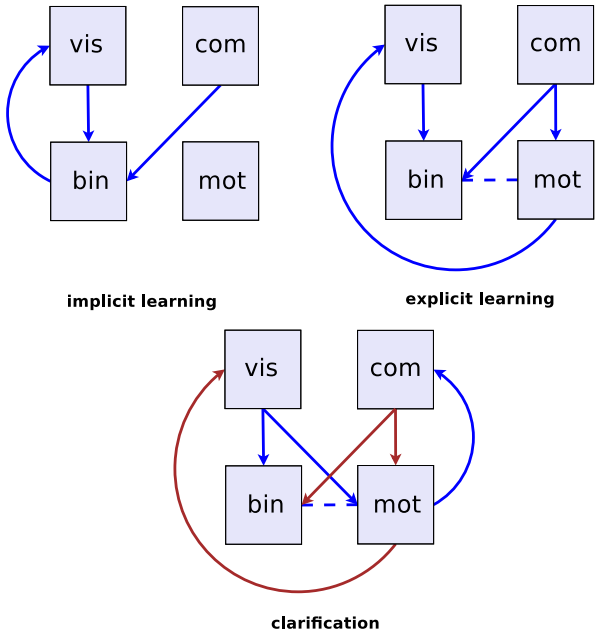

Fig. 4. The flow of information between subsystems (vision, communication, binding, motivation) in cross-modal learning mechanisms. The dashed line represents information referencing, while the red arrows represent the reaction to the clarification request.

to improve its services to other modalities and adapt itself to the changing environment, the subsystem has to make good use of such feedback, since it represents the most important guideline for further interpretation of the sensorial input. The necessity for visual learning is therefore the result of the needs for adaptation to both environments: external environment, which is perceived through visual sensors, and internal environment — composed by other cognitive subsystems.

Our system currently bases its visual learning on a dialogue with a human tutor. This means that the visual subsystem organizes its knowledge according to the tutor's interpretation of the visual environment. Such knowledge is therefore composed of associations between internal representations of human semantic categories and representations of visual sensorial input. In the same manner other types of cross-modal knowledge associations are possible. The learning is achieved via two distinct learning mechanisms: *explicit learning* and *implicit learning*.

The *explicit learning* is a pure tutor-driven learning mechanism. It handles situations when tutor explicitly expresses himself about a certain object property, e.g. when the main purpose of his communicative act was to provide a novel information to the system. The *implicit learning*, on the other hand, is triggered by system's own initiative when it recognizes a learning opportunity in the current situation, usually exploiting certain favorable circumstances. In our current system, for example, the information which primary purpose was the identification of an entity and the binding of its multi-modal representations, can be reused for updating visual concepts. Implicit learning is usually used to improve

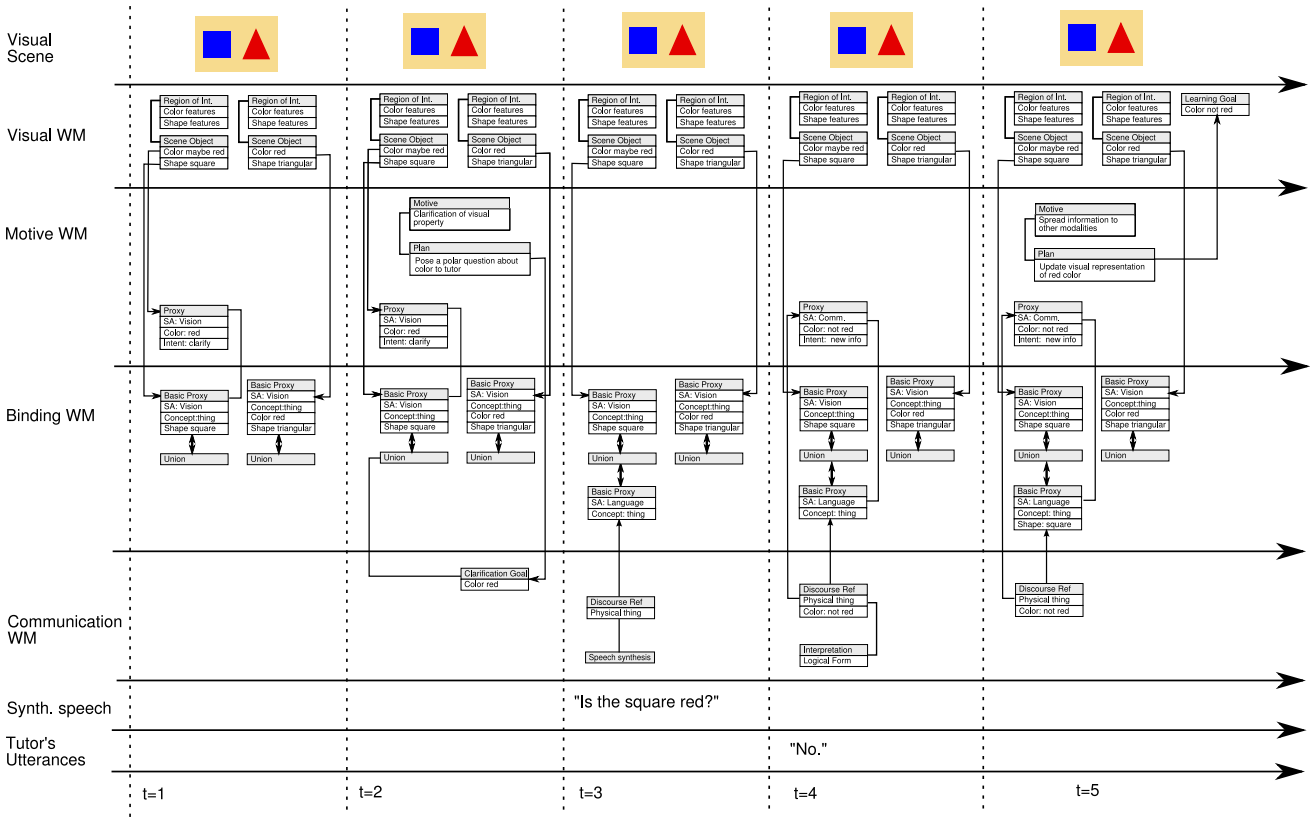

Fig. 5. An example of clarification mechanism (time-plot). Boxes represent data structures in the working memories (WM) of various subarchitectures. Directed edges connecting the boxes show the flow of information, while undirected edges denote a close relation or referencing between the connected structures.

already known concepts: to fill their knowledge gaps and to raise (or lower) the system's confidence in them. Explicit learning on the other hand is essential for learning new concepts or to radically alter old ones (e.g. unlearning). In this case the information to be learned is never used for identification and binding, since it can jeopardize both processes, if the system's beliefs are not correct. The information is passed to visual learner as an explicit learning motivation which references the a-modal representation of the corresponding entity.

In both mechanisms the communication subsystem analyzes the tutor's utterance, forms adequate representations of described concepts and exports them to the Binding SA as binding proxies, so that each referenced entity is represented by its own proxy. Explicitly expressed properties — properties that are perceived not as identification cues, but rather as a focal points of the message or even as a learning instructions — are not sent to Binding SA, the interface layer forwards them to the Motivation SA instead. The proxy in Motivation SA (motivation proxy) also contains a reference to the binding proxy and the motive to spread its information across the system. In very similar fashion, the visual subsystem exports its own binding proxies. Visual binding proxies represent segmented objects with their recognized attributes. Through a bound proxy the visual binding monitor gains access to linguistic information in the binding union, which it can associate with the visual features of a particular scene object, thus implicitly gaining a labeled learning example.

In the case of explicit learning the planning subsystem, using the information in the motivation proxy, makes a plan which results in learning instruction in Vision SA. Besides a linguistic label the learning instruction also contains a reference to the binding union representing the scene object in question. And that again leads to the object's visual features, which are used to update the current internal representations.

3) Clarification: The clarification mechanism is a means for cross-modal verification of modal information. It is typically used when a component is not very confident about certain recognition or prediction outcome. Instead of completely rejecting it, the motivation monitor creates a motivation proxy containing the unreliable information together with the clarification instruction and a reference to the binding proxy representing the scene object. Depending on the available plans and resources the clarification request results in a specific action within another modality, which helps the system to acquire additional information. The clarifying action always involves the entity represented by the referenced binding union. Often that action would be a polar question about the certain object property synthesized by the communication subsystem. The clarifying action usually triggers some kind of reaction, where the information flows in the opposite direction (e.g. the interpretation of the tutor's

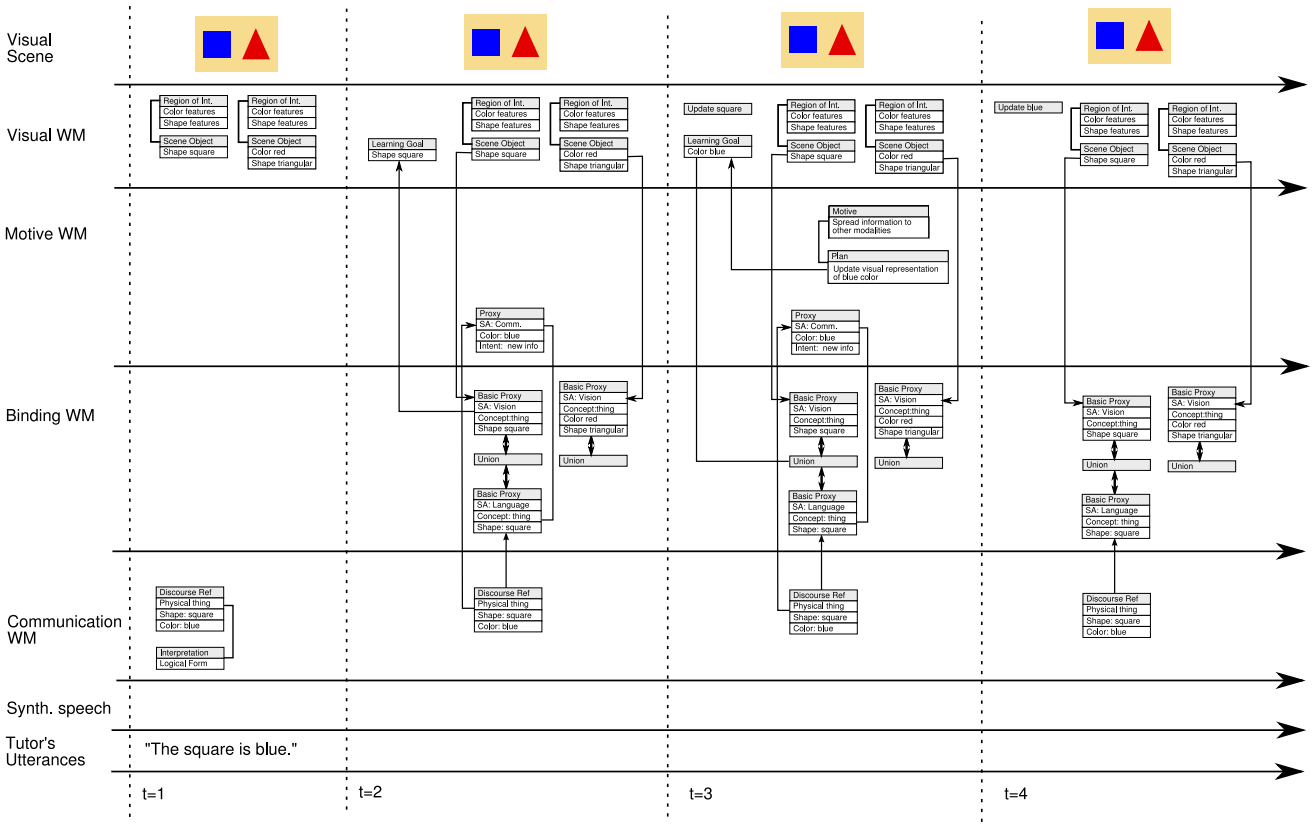

Fig. 6. An example of cross-modal interaction including implicit and explicit learning mechanisms (time-plot). Boxes represent data structures in the working memories (WM) of various subarchitectures. Directed edges connecting the boxes show the flow of information, while undirected edges denote a close relation or referencing between the connected structures.

answer).

IV. Examples of Cross-modal Interaction

We will illustrate the mechanisms described in the previous section with examples of clarification and visual learning. Both examples are table-top scenarios. They occur within the same scene consisting of simple objects on the table: a blue square and a red triangle. The examples assume that the system's model for the red color is too inclusive, so that it contains also parts of the blue space, while the model for the blue is missing (the system has yet to learn it). The color of the square object on the table is thus deemed red, but the recognition confidence is low.

A. Clarification Example

The clarification request is a reaction to the low recognition confidence for the square object's color. The visual subsystem seeks cross-modal verification for its red color model, which occurs in the form of a polar question to the tutor. The system reacts to the tutor's reply by unlearning the red color label on square object's visual features.

Figure 5 shows the clarification flow. The scene processing in Visual SA results in two working memory entries per object. The *ROI (Region of Interest)* WM entries represent the output of the *quantitative* scene processing, while the *qualitative* layer stores its recognition results in the *SceneObject* WM entries. For each SceneObject WM entry the Visual Binding Monitor creates a binding proxy. The Binding SA reacts to the proxies by assigning them to the binding unions (in this case a new union is created for each proxy, since there are no other unions they could be related to). In the case of the blue square the Learner-recognizer is not confident in color recognition result (red), therefore the color property is not included into the binding proxy. Instead, the *Motivation Monitor* creates a motivation proxy seeking clarification from other modalities about the object's color (timeframe 1). The motivation proxy references the object's binding proxy. Based on the motivation proxy, the Motivation SA generates a *motive* and then a *plan* how to get the missing information. The plan suggests that the Communication SA could best handle the request, therefore a *clarification goal* is created in the Communication SA (timeframe 2). The goal contains unreliable information on the one hand and a reference to the square object's binding union on the other, which enables the Communication SA to generate a polar question about the object's color. The question's WM entry (discourse referent) has its own binding proxy that, due to the reference provided by the clarification goal, binds directly into the object's union (timeframe 3).

After the tutor answers the polar question, a similar process is performed in the opposite direction. The Communication SA reacts to the tutor's answer by creating a *'new information'* motivation proxy (timeframe 4). Using the

same mechanisms as in previous steps, the system creates a learning goal in Vision SA (timeframe 5), which eventually results in an update of the model — unlearning in this case.

B. Visual Learning Example

The visual learning in the second example is a direct consequence of the clarification request. The tutor tries to explicate his previous answer by specifying the object's true color: "The square thing is blue". In this sentence the shape property (square) is used to identify the object which the tutor is referring to, when explicitly expressing the color property. This utterance triggers both, explicit (color) and implicit learning (shape). While explicit learning directly and arbitrarily fulfills the user's expectations, the implicit learning is more autonomous and incidental.

As we can see in Figure 6 it is the Communication SA that in this case splits in two parts the information about the square objects. The implicit information part goes to the binding proxy and it is used by the Binding SA to group the corresponding visual and communication proxies into a single binding union, thus relating the tutor's message to the referred visual object. The Visual Binding Monitor derives the implicit learning goal directly from the Binding SA by comparing the binding union to the visual proxy (timeframe 2). The explicit part follows a similar path to the one already seen in the clarification example: a motivation proxy is submitted, containing the explicit information and a reference to the object's binding proxy (timeframe 2). After that the motivation and planning mechanisms create an explicit learning goal in Vision SA (timeframes 3 and 4).

From these examples we can clearly see why it is important to separate explicit and implicit learning paths. If the explicitly expressed color property was communicated through a binding proxy, it would be also used for the identification. In our case this could jeopardize the binding process. Despite unlearning, the square object's color could be still recognized as red (as in the example in Figure 6), which would prevent the linguistic and visual proxies to bind to a common union.

V. Conclusion

In this paper we presented a generic method for integrating visual components into a multi-modal cognitive system based on CoSy architecture schema. We described the visual part of one possible instantiation and its cross-modal interaction. The visual subsystem emphasizes the visual learning through communication with a human tutor. We exemplified the cross-modal integration with clarification mechanism and mechanisms for implicit and explicit learning.

Our future research will be focused on further improvement of our visual instantiations. We will improve the attention mechanisms and extend and robustify the object detection and recognition methods as well as the learning methods. We aim to support a mobile robot platform and extend the visual learning mechanisms with the capabilities for self-reflection and detection of ignorance.

In the paper we have also shown the system's capability to form a-modal shared representations of individual entities. Its ability to extend cross-modal concepts is currently quite limited, however. The cross-modal self-extension ability of the integrated system will also be an important topic of our future research.

On the architecture level, the architecture schema will undergo several changes that will enable robust and efficient behavior and reliable self-extension of the cognitive system.

VI. Acknowledgements

This research has been supported in part by the EU FP7 project CogX (ICT-215181) and Research program Computer Vision (RS).

References

[1] The PASCAL object recognition database collection. *http://pascallin.ecs.soton.ac.uk/challenges/VOC/databases.html*.

[2] E. Ardizzone, A. Chella, M. Frixione, and S. Gaglio. Integrating subsymbolic and symbolic processing in artificial vision. *Journal of Intelligent Systems*, 1(4):273–308, 1992.

[3] B. Bolder, H. Brandl, M. Heracles, H. Janssen, I. Mikhailova, J. Schmüdderich, and C. Goerick. Expectation-driven autonomous learning and interaction system. In *IEEE-RAS International Conference on Humanoid Robots*, to appear 2008.

[4] M. Brenner, N. Hawes, J. Kelleher, and J. Wyatt. Mediating between qualitative and quantitative representations for task-orientated human-robot interaction. In *Proceedings of the Twentieth International Joint Conference on Artificial Intelligence (IJCAI)*, Hyderabad, India, 2007.

[5] A. Chella, M. Frixione, and S. Gaglio. A cognitive architecture for artificial vision. *Artif. Intell.*, 89(1-2):73–111, 1997.

[6] H. I. Christensen, A. Sloman, G-J. Kruijff, and J. Wyatt, editors. *Cognitive Systems*. http://cognitivesystems.org/cosybook/index.asp, 2009.

[7] L. D. Erman, F. Hayes-Roth, V. R. Lesser, and D. R. Reddy. The hearsay-ii speech understanding system: integrating knowledge to resolve uncertainty. *Computing Surveys*, 12:213–253, 1980.

[8] S. Harnad. The symbol grounding problem. *Physica D*, 42(1-3):335–346, June 1990.

[9] N. Hawes, A. Sloman, and J. Wyatt. Towards an empirical exploration of design space. In *Evaluating Architectures for Intelligence: Papers from the 2007 AAAI Workshop*, pages 31 – 35, Vancouver, Canada, July 2007.

[10] Nick Hawes, Aaron Sloman, Jeremy Wyatt, Michael Zillich, Henrik Jacobsson, Geert-Jan Kruijff, Michael Brenner, Gregor Berginc, and Danijel Skočaj. Towards an integrated robot with multiple cognitive functions. In *AAAI*, pages 1548–1553. AAAI Press, 2007.

[11] Nick Hawes, Michael Zillich, and Jeremy Wyatt. BALT & CAST: Middleware for cognitive robotics. In *Proceedings of IEEE RO-MAN 2007*, pages 998 – 1003, August 2007.

[12] H. Jacobsson, N. Hawes, G.-J. Kruijff, and J. Wyatt. Crossmodal content binding in information-processing architectures. In *HRI '08: Proceedings of the 3rd ACM/IEEE international conference on Human robot interaction*, pages 81–88, New York, NY, USA, 2008. ACM.

[13] H. Jacobsson, N. Hawes, D. Skočaj, and G.-J. Kruijff. Interactive learning and cross-modal binding - a combined approach. In *Symposium on Language and Robots*, Aveiro, Portugal, 2007.

[14] M. Kristan, D. Skočaj, and A. Leonardis. Incremental learning with Gaussian mixture models. In *Computer Vision Winter Workshop*, pages 25–32, 2008.

[15] D. K. Roy. Learning visually-grounded words and syntax for a scene description task. *Computer Speech and Language*, 16(3-4):353–385, 2002.

[16] D. K. Roy and A. P. Pentland. Learning words from sights and sounds: a computational model. *Cognitive Science*, 26(1):113–146, 2002.

[17] D. Skočaj, M. Kristan, and A. Leonardis. Continuous learning of simple visual concepts using Incremental Kernel Density Estimation. In *International Conference on Computer Vision Theory and Applications*, pages 598–604, 2008.